AN ITERATIVE LEARNING CONTROL APPROACH TO IMPROVING FIDELITY IN INTERNET-DISTRIBUTED HARDWARE-IN-THE-LOOP SIMULATION

Tulga Ersal
Department of Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109
tersal@umich.edu

Mark Brudnak
The US Army Tank-Automotive Research,
Development and Engineering Center
Warren, MI, 48397
mark.j.brudnak.civ@mail.mil

Jeffrey L. Stein
Department of Mechanical Engineering
University of Michigan
Ann Arbor, MI 48109
stein@umich.edu

ABSTRACT

One of the main challenges of co-simulating hardware-in-the-loop systems in real-time over the Internet is the fidelity of the simulation. The dynamics of the Internet may significantly distort the dynamics of the network-integrated system. This paper presents the development of an iterative learning control based approach to improve fidelity of such networked system integration. Towards this end, a new metric for characterizing fidelity is proposed first, which, unlike some existing metrics, does not require knowledge about the reference dynamics (i.e., dynamics that would be observed, if the system was physically connected). Next, using this metric, the problem of improving fidelity is formulated as an iterative learning control problem. Finally, the proposed approach is illustrated on a purely simulation-based case study. The conclusion is that the proposed approach holds significant potential for achieving high fidelity levels.

Keywords: real-time hardware-in-the-loop simulation, networked simulation, fidelity, iterative learning control

I. INTRODUCTION

The key benefit of hardware-in-the-loop (HIL) simulation is well known: it uniquely combines the advantages of physical prototyping and simulation-based engineering and thus allows for experiments that are at the same time cost effective and highly accurate [1]. It has therefore become indispensable in many application areas, such as automotive [2, 3], aerospace [4, 5], manufacturing [6], robotics [7, 8], and defense [9, 10].

Recently, Internet-distributed HIL simulation (ID-HIL) started attracting interest as a framework that enables concurrent system engineering even if the components that comprise the desired HIL setup are geographically distributed. This idea has found applications in the fields of earthquake engineering [11-16] and automotive engineering [17-23], and is also closely related to the teleoperation idea in robotics and haptics [24-31].

One of the key challenges in ID-HIL simulation is ensuring fidelity. In this context, fidelity refers specifically to how close the dynamics of the networked system are to the dynamics that would be observed if the system was physically integrated (i.e., reference dynamics). To characterize fidelity, the literature has proposed several methods. For example, a frequency-domain metric called distortion was proposed as the normalized difference between the networked and reference dynamics [32, 33]. In addition, the telerobotics and haptics literatures have proposed many other frequency domain metrics, which could be adopted into the ID-HIL framework, as well [34-39]. A time-domain, statistical approach was also proposed to distinguish the inherent variation in the reference dynamics from the additional variation introduced due to the network [20, 23].

To improve fidelity, the literature proposes several methods, as well. These methods range from selecting appropriate coupling points [32, 33] to using feedback control [40-42] and observer-based approaches [17-19]. The limitations

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **15 JUN 2012**	2. REPORT TYPE **Journal Article**	3. DATES COVERED **21-05-2012 to 14-06-2012**

4. TITLE AND SUBTITLE **AN ITERATIVE LEARNING CONTROL APPROACH TO IMPROVING FIDELITY IN INTERNET-DISTRIBUTED HARDWARE-IN-THE-LOOP SIMULATION**	5a. CONTRACT NUMBER **W56Hzv-04-2-0001**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Tulga Ersal; Mark Brudnak; Jeffrey Stein**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Mechanical Engineering,University of Michigan,1600 Huron Parkway,Ann Arbor,MI,48109**	8. PERFORMING ORGANIZATION REPORT NUMBER **; #23021**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **U.S. Army TARDEC, 6501 E.11 Mile Rd, Warren, MI, 48397-5000**	10. SPONSOR/MONITOR'S ACRONYM(S) **TARDEC**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) **#23021**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Paper submitted to Dynamic System and Control Conference.

14. ABSTRACT

One of the main challenges of co-simulating hardware-in-the-loop systems in real-time over the Internet is the fidelity of the simulation. The dynamics of the Internet may significantly distort the dynamics of the network-integrated system. This paper presents the development of an iterative learning control based approach to improve fidelity of such networked system integration. Towards this end, a new metric for characterizing fidelity is proposed first, which, unlike some existing metrics, does not require knowledge about the reference dynamics (i.e., dynamics that would be observed, if the system was physically connected). #ext, using this metric, the problem of improving fidelity is formulated as an iterative learning control problem. Finally, the proposed approach is illustrated on a purely simulation-based case study. The conclusion is that the proposed approach holds significant potential for achieving high fidelity levels.

15. SUBJECT TERMS

real-time hardware-in-the-loop simulation, networked simulation, fidelity, iterative learning control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Public Release**	**10**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

of these techniques can be briefly summarized as follows. Coupling-point-based approaches work well, if a coupling point with desired characteristics exists. However, such a coupling point may not always exist. Feedback-based techniques are subject to well-known fundamental trade-offs [41-43]; i.e., improving fidelity at one frequency compromises it at another frequency. Observer-based techniques rely on the existence of high-fidelity models. However, not having such models or avoiding the need to develop such models is the main motivation behind the HIL paradigm. Thus, improving fidelity in ID-HIL systems is still an open research question and is the focus of this paper.

In an effort to overcome the abovementioned limitations of the existing techniques, this paper proposes an iterative learning based approach to improving fidelity in ID-HIL systems. First, a new metric to characterize fidelity is proposed. Unlike some of the existing metrics, e.g. [20, 23, 32, 33], this metric does not require knowledge about the reference dynamics to quantify fidelity. Next, the paper formulates the fidelity-improvement problem as an iterative learning control problem. Finally, the proposed formulation is applied to a simulation-based case study that highlights the potential performance of the proposed approach.

II. CHARACTERIZING FIDELITY

As mentioned in Section I, many metrics have been proposed to characterize fidelity, especially in telerobotics and haptic systems. There are two fundamental barriers to adopting the existing metrics to the ID-HIL framework: (1) Most of these metrics have been developed in a linear and deterministic framework. However, ID-HIL systems are, in general, nonlinear and stochastic. (2) Most of the existing metrics assume that the desired system dynamics are known and utilize this knowledge as a benchmark to measure the fidelity of the networked system. The benchmark for an ID-HIL system would be a physical assembly of the system. However, the main motivation of the ID-HIL paradigm is the lack of availability of a physical assembly. Thus, even though the existing metrics could still be helpful in a research environment where a physical assembly could be made available to develop and test ID-HIL techniques, in practice the desired system dynamics are unknown and most of the existing fidelity metrics cannot be defined.

Therefore, the aim of this section is to characterize fidelity in an ID-HIL system despite the lack of knowledge about the dynamics that the ID-HIL system is to emulate as accurately as possible. Towards this end, this work uses the error between the instantaneous values of coupling signals at both ends of the network.

To explain this idea on an example, consider the ID-HIL framework shown in Fig. 1. This is a two-site ID-HIL example, in which Systems 1 and 2, both of which may include hardware and models, are integrated over a network through the coupling variables c_1 and c_2. Due to the numerous considerations associated with integration over a network (e.g., network delays, communication bandwidth, filters, sampling, etc.), the instantaneous value of the coupling variable i will not be the same on the two sides of the network. Let $c_{i,j}(t)$ represent the value of c_i as seen by System j at time t. Ideally, i.e., if the systems were co-located and coupled physically, we would have $c_{i,1}(t) = c_{i,2}(t)$. In the ID-HIL case, however, the two instances of the variable are not the same, i.e., $c_{i,1}(t) \neq c_{i,2}(t)$. Thus, we propose to use the difference between $c_{i,1}$ and $c_{i,2}$ as a metric for fidelity in the coupling variable i; i.e.,

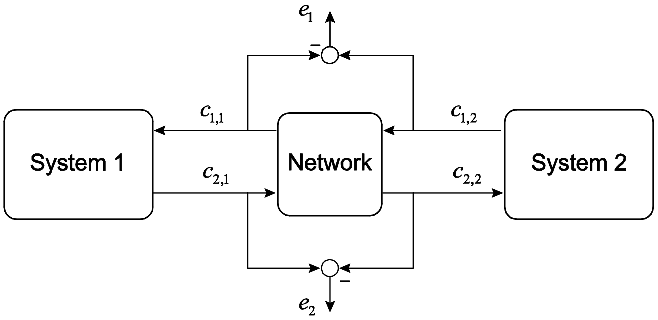

FIG. 1. ILLUSTRATION OF DEFINING THE ERRORS IN THE COUPLING SIGNALS TOWARDS CHARACTERIZING FIDELITY IN ID-HIL ON AN EXAMPLE WITH TWO SITES (SYSTEM 1 AND 2). $c_{i,j}$ REPRESENTS THE *I*-TH COUPLING VARIABLE ON THE SYSTEM *J* SIDE OF THE NETWORK, AND e_i REPRESENTS THE INSTANTANEOUS ERROR IN THE *I*-TH COUPLING VARIABLE.

$$\begin{aligned} e_1(t) &= c_{1,2}(t) - c_{1,1}(t), \\ e_2(t) &= c_{2,1}(t) - c_{2,2}(t). \end{aligned} \tag{1}$$

Note that causality is taken into account while defining the error signals in (1) when using the system output instances of the variables (i.e., $c_{1,2}$ and $c_{2,1}$) as the references for the system input instances of the same variables (i.e., $c_{1,1}$ and $c_{2,2}$).

It must also be emphasized that increasing fidelity of one of the coupling signals does not automatically imply an increase in the fidelity of the remaining coupling signals, as shown in [32, 33]. Thus, in general, no single coupling signal error can completely describe the system fidelity by itself. Therefore, to increase the *system* fidelity, this error needs to be driven to zero for *all* coupling variables. With respect to Fig. 1, for example, this implies $e_1 := c_{1,2} - c_{1,1} \to 0$ and $e_2 := c_{2,1} - c_{2,2} \to 0$.

To characterize *system* fidelity, all coupling signal errors can be aggregated into a single error metric. One way to achieve this is to use the following norm of weighted error norms:

$$E := \sqrt{\sum_{i=1}^{n} \left(w_i \|e_i\|_2 \right)^2} . \tag{2}$$

where n is the total number of coupling signals. The goal of maximizing the system fidelity then translates to minimizing the system-level error metric E.

The rationale behind using different weights for different error vectors is that the error in one coupling signal may be more critical than the error in another for a particular output of interest. In this paper, we will consider the weight

$$w_i = \frac{1}{\left\| e_i^0 \right\|_2}; \tag{3}$$

i.e., the norm of error of each coupling variable will be normalized with respect to its own initial value.

III. IMPROVING FIDELITY USING ITERATIVE LEARNING CONTROL

The fundamental problem with the proposed fidelity metric is that it is not available online; i.e., the value of $c_{i,1}$ and $c_{i,2}$ at a given instant cannot be made available for the error calculation at the same instant. The error $e_i(t)$ can be known only at $t+\tau$, i.e., after some delay τ that is needed to collocate the measurements $c_{i,1}(t)$ and $c_{i,2}(t)$. Thus, the error cannot be used without delay to make online corrections to improve fidelity while an experiment is running. Even though the signal could be made available with a delay, so that a feedback loop could be closed around it, since the overarching goal of this effort is to go beyond the fundamental limitations of such feedback methods, this option will not be discussed any further in this paper.

Instead, this work considers an offline application of the described metric to improve fidelity. To achieve this, the iterative learning control (ILC) paradigm is leveraged to improve fidelity of an ID-HIL experiment iteratively. The rest of this section describes this framework.

The ILC-based framework used in this work is illustrated in Fig. 2 for one of the coupling signals as an example. The error as defined in Eq. (1) is provided offline as the error signal to the ILC algorithm. Together with the control input used in the corresponding run, the algorithm then shapes the control input according to a learning algorithm to attenuate the error in the next run, i.e.,

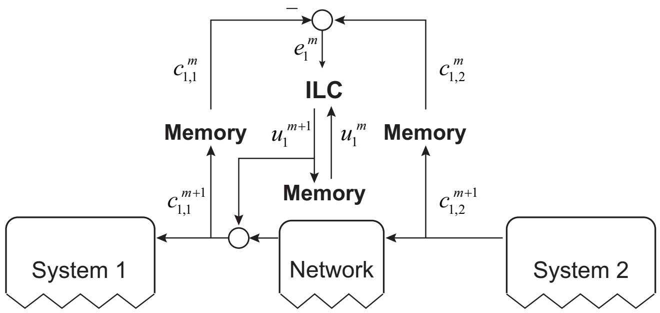

FIG. 2. PROPOSED ILC-BASED FRAMEWORK FOR ITERATIVELY IMPROVING ID-HIL FIDELITY. THIS FIGURE ILLUSTRATES A DECENTRALIZED APPROACH IN WHICH AN INDEPENDENT ILC CONTROLLER IS UTILIZED FOR EACH COUPLING SIGNAL.

$$\mathbf{u}_i^{m+1} = \mathbf{f}\left(\mathbf{u}_i^m, \mathbf{e}_i^m\right), \tag{4}$$

with

$$\begin{aligned} \mathbf{u}_i^m &= \left[u_i^m(0) \quad u_i^m(1) \quad \cdots \quad u_i^m(N) \right]^T, \\ \mathbf{e}_i^m &= \left[e_i^j(0) \quad e_i^m(1) \quad \cdots \quad e_i^m(N) \right]^T, \end{aligned} \tag{5}$$

where the superscript m is an index for the iteration, and N represents the number of time steps.

It is very important to emphasize here that this problem is not as trivial as using the error from the initial run and adding it to the coupling variable in the current run. The reason is the bidirectional nature of the coupling between the systems, which prevents the applicability of this trivial solution. For example, referring to the example in Fig. 2, any modification to $c_{1,1}^{m+1}(k)$ will propagate through System 1, Network, and System 2 and affect $c_{1,2}^{m+1}(n)$, $n = k+1, k+2, \ldots, N$. In other words, if $c_{1,2}$ is regarded as a reference for $c_{1,1}$ in the ILC framework, then the problem can be considered as the reference signal changing as a result of the ILC action.

Thus, designing the learning function $\mathbf{f}\left(\mathbf{u}_i^m, \mathbf{e}_i^m\right)$ properly is crucial. Different learning functions could be considered for their suitability to ID-HIL. In this paper, the ILC learning algorithm of the form

$$u_i^{m+1}(k) = Q(q)\left(u_i^m(k) + L(q) e_i^m(k+1)\right) \tag{6}$$

is considered, since it was reported to be a widely used algorithm [44]. In (6), q represents the forward time shift operator. Due to their applicability to nonlinear systems and wide use [44], PD-type learning functions are considered here, e.g.,

$$u_i^{m+1}(k) = u_i^m(k) + k_p e_i^m(k) + k_d\left(e_i^m(k) - e_i^m(k-1)\right) \tag{7}$$

This type of learning functions also has the advantage of not requiring a model of the system as part of the design process [45-54], which is particularly suitable for the ID-HIL paradigm and the specific goals of this paper.

This ILC-based framework is applied to each coupling signal independently. In this framework, each coupling signal has its own ILC controller (with its own learning function) that does not communicate with the ILC controllers of the remaining coupling signals. Through such a decentralized approach, scalability could be achieved; i.e., the method would be easy to extend to ID-HIL setups with multiple sites.

IV. CASE STUDY

To investigate the viability of the framework described in Sections II and III, a completely simulation-based study is performed with a vehicle and driver system. A high-level overview of the system is shown in Fig. 3. The major

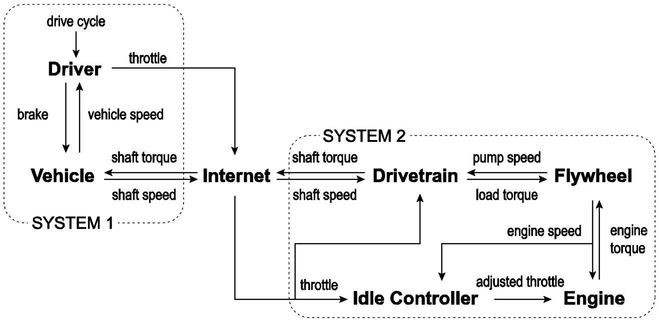

FIG. 3. THE OUTLINE OF THE SYSTEM CONSIDERED IN THE PRELIMINARY STUDY

components are the Driver, Vehicle, Internet, Drivetrain, Flywheel, Idle Controller, and Engine. The Driver and Vehicle subsystems constitute System 1, whereas System 2 comprises the Drivetrain, Flywheel, Engine, and Idle Controller. The two Systems are connected over the Internet through the coupling signals throttle, transmission shaft torque and transmission shaft speed. Hence, three instances of the ILC framework shown in Fig. 2 are employed for the three coupling signals.

This section first gives the details of these components and then presents the results for the ILC-based fidelity improvement.

System Description

The driver model is a PI controller with saturation and anti-windup. It takes the difference between the desired and actual vehicle velocities as inputs and generates an output within the interval $[-1,1]$, where positive and negative values correspond to throttle and brake commands, respectively.

The vehicle dynamics model is a point mass representation of a military vehicle (the High Mobility Multipurpose Wheeled Vehicle - HMMWV) and includes differentials, wheel inertia, a Coulomb and viscous friction based brake model, rolling resistance, aerodynamic drag, and tire slip. The model can be expressed in the following form

$$\left(m_{vehicle}+\frac{J_{wheel}}{r_{wheel}^{2}}\right)\ddot{x}=\left(\tau_{differential}-b\frac{\dot{x}}{r_{wheel}}-\tau_{brake}\right)\frac{1}{r_{wheel}} -f_{rolling}-f_{aero} \qquad (8)$$

where $m_{vehicle}$ is vehicle mass, J_{wheel} is wheel inertia, b is wheel viscous damping coefficient, r_{wheel} is wheel radius, and $\ddot{x}$ is vehicle acceleration. Furthermore, f_{aero} and $f_{rolling}$ are aerodynamic and rolling resistance forces, respectively, and $\tau_{differential}$ and τ_{brake} are differential and brake torques, respectively, and they are given by

$$f_{aero}=\frac{1}{2}A\rho_{air}C_d\left|\dot{x}\right|\dot{x} \qquad (9)$$

where A is the vehicle frontal area, ρ_{air} is the air density, C_d is the aerodynamic drag coefficient, and $\dot{x}$ is vehicle velocity;

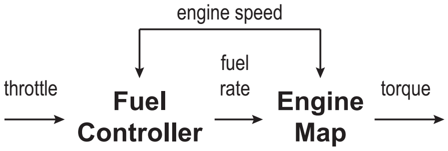

FIG. 4. THE ENGINE MODEL.

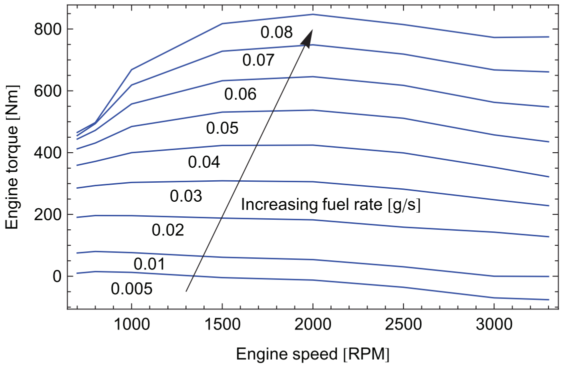

FIG. 5. THE STATIC ENGINE MAP.

$$f_{rolling}=sgn\left(\dot{x}\right)\left(a_0+a_1F_z+a_2\frac{F_z}{P}+a_3\frac{F_z^2}{P}\right) \qquad (10)$$

where a_i are empirical coefficients, and P is the tire pressure; and

$$\tau_{brake}=\left(b_2\frac{\dot{x}}{r_{wheel}}+sgn\left(\frac{\dot{x}}{r_{wheel}}\right)F_{Coulomb}\right)c_{brake} \qquad (11)$$

where b_2 is brake viscous damping coefficient, $F_{Coulomb}$ is the static Coulomb force, and c_{brake} is the brake command from the driver.

A map-based engine model is considered as shown in Fig. 4. The map is a static map obtained experimentally from a physical 6L V8 diesel engine. The input to the map is the fuel rate and the engine speed, and the output is the engine torque. Fig. 5 shows the engine map.

The fuel rate is given by the fuel controller, which fulfills two purposes. First, it implements the turbo lag as a first order system of the form $k/\left(\tau s+1\right)$. The input to the turbo lag is the difference between the maximum fuel rate for the given engine RPM and the naturally aspirated fuel rate, both determined experimentally. The output of the turbo lag is added to the naturally aspirated fuel rate. This sum is then compared to the maximum possible fuel rate for the given engine RPM and throttle, and the minimum of the two is taken as the unadjusted fuel rate. Second, the fuel controller monitors the unadjusted fuel rate and adjusts it if the engine speed falls below 650 RPM or exceeds the maximum rated speed of 3300 RPM to bring the speed back to the desired operating region.

The idle controller is a PI controller with saturation and anti-windup, and is activated when the throttle demand from the driver falls below 11% to maintain an engine idle speed of 750 RPM.

The flywheel model is an inertia element that takes the load torque and the engine torque as inputs and determines the pump/engine speed as the output. This speed signal is fed to the drivetrain model as the pump speed, as well as to the engine and idle controller as the engine speed.

The drivetrain model includes the torque converter, transmission, and shift logic. The torque converter model is a static model that takes pump and turbine speeds as inputs and generates pump and turbine torques according to the equations

$$\begin{aligned} \tau_{pump} &= \left(\frac{\omega_{pump}}{\kappa(\omega_r)} \right)^2 sign(1-\omega_r) \\ \tau_{turbine} &= \alpha(\omega_r)\tau_{pump} \end{aligned} \qquad (12)$$

where $\omega_r = \omega_{turbine} / \omega_{pump}$ is the speed ratio between turbine and pump speeds, $\kappa(\omega_r)$ is a piecewise function approximating a desired capacity factor curve, and $\alpha(\omega_r)$ is a piecewise linear function approximating a desired torque ratio curve. The pump side of the torque converter is connected to the engine flywheel; thus, pump speed and engine speed are the same. The turbine side of the torque converter, on the other hand, is connected to the transmission model.

The transmission model takes into account the transmission shaft inertia, stiffness, and damping, as well as the gear inefficiencies and torque losses due to fluid churning. Specifically, the speed reduction in each gear is assumed to be ideal, while the torque multiplication is assumed to be scaled by an efficiency factor. Furthermore, the torque lost due to fluid churning is modeled as variable nonlinear resistance of the form

$$\tau_{chruning\ loss} = r_1(gear)\omega_{shaft} + r_2(gear)\omega_{shaft}^2 \qquad (13)$$

where r_1 and r_2 are coefficients that change depending on the gear.

The inputs to the shift logic, the final element in the drivetrain model, are the transmission output shaft speed and the throttle demanded by the driver. The simple chart shown in Fig. 6 is used to determine if a shift is to be initiated. The solid and dashed lines in Fig. 6 indicate upshift and downshift thresholds, respectively. Note that this chart is only a crude approximation of a real shift map, but it is employed here for simplicity.

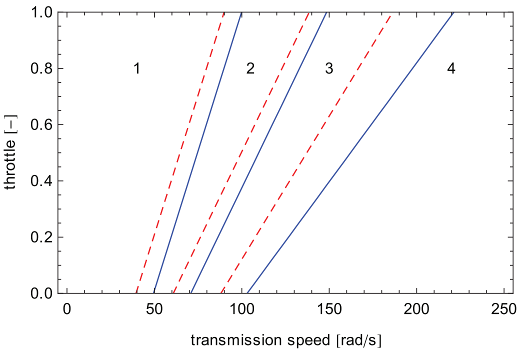

FIG. 6. GEARSHIFT LOGIC.

Modeling the Networking of the Systems

This section describes the architecture used for simulating the communication of the coupling variables over the Internet between System 1 and 2. In the adopted framework, the System 1 acts as the client and the System 2 acts as the server. System 1 sends updated transmission speed and throttle signals at a frequency of 20 Hz, regardless of whether it receives a response or not. System 2, on the other hand, only responds to the packets it receives, i.e., it only sends an updated transmission torque signal when it receives a packet from System 1. Fig. 7 and 8 illustrate the communication flowcharts for System 1 and 2, respectively.

The Internet is modeled to allow for an optimization based tuning of the ILC parameters, as well as to avoid clock synchronization issues between the two Systems.

To develop a representative model of the Internet, the network between two computers in Ann Arbor, MI, and Warren, MI, was characterized through a series of experiments, in which packets ranging from 64 bytes to 1024 bytes were exchanged

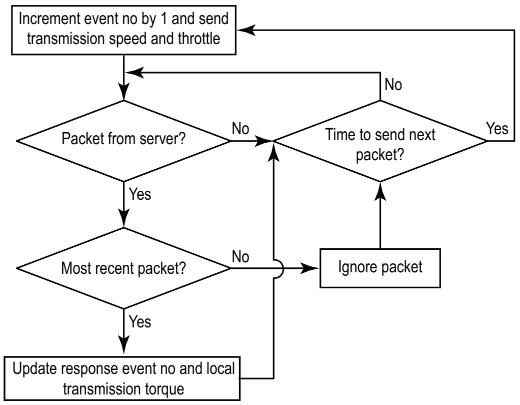
Increment event no by 1 and send transmission speed and throttle
No
Packet from server?
No
Time to send next packet?
Yes
Yes
Most recent packet?
No
Ignore packet
Yes
Update response event no and local transmission torque

FIG. 7. COMMUNICATION FLOWCHART FOR THE SYSTEM 1

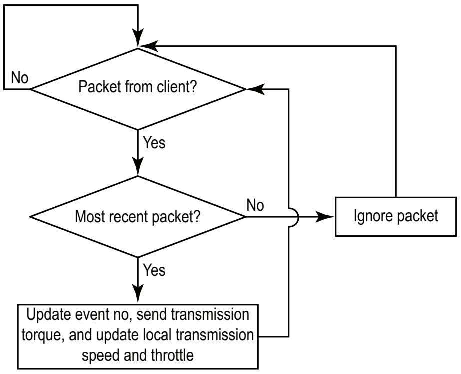
No
Packet from client?
Yes
Most recent packet?
No
Ignore packet
Yes
Update event no, send transmission torque, and update local transmission speed and throttle

FIG. 8. COMMUNICATION FLOWCHART FOR THE SYSTEM 2

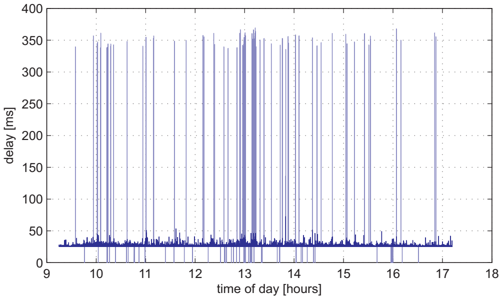

FIG. 9. CHARACTERIZATION OF NETWORK QUALITY OF SERVICE BETWEEN TWO COMPUTERS IN ANN ARBOR, MI, AND WARREN, MI.

TABLE 1. STATISTICS FOR THE RESULTS IN FIG. 9

Number of packets	1,110,000
Packet size	1024 B
Avg. delay	25.7 ms
Min delay	25.3 ms
No of spikes	114 (0.01%)
No of drops	56 (0.005%)

on different days and at different times of day using the UDP/IP protocol. This protocol is preferred for its speed, but does not guarantee packet delivery. A typical result for round trip time delay vs. time of day obtained from one those experiments is shown in Fig. 9. The figure clearly shows a multi-modal character in the sense that some packets experience a delay around 25 ms, and some around 350ms (spikes), while others are dropped (shown as zero delay in the figure). A packet is considered dropped in this case if it does not arrive within 1s. Table 1 provides some statistics of the results shown in Fig. 9.

Based on the statistics given in Table 1, the percentages of spikes and dropped packets are very small. Thus, as a first approximation, the spikes and drops are neglected, and the emphasis is given to the characterization of the dominant mode of the delay, which appears in the darker potion of the curve in Fig. 9 around 25ms delay.

For the purposes of this study, the delay is assumed to be an independent random variable. The assumption of independence is validated by checking the autocorrelation of the observed delay sequence after the average delay is subtracted (Fig. 10). The fact that the autocorrelation is almost zero for all lags except for zero indicates that there is no correlation between the delay values, thereby justifying the assumption of independence.

The histogram of the dominant mode of delay is shown in Fig. 11. The figure also shows that the dominant mode can be approximated quite accurately with a lognormal distribution with parameters $\mu_{rt} = -0.9$ and $\sigma_{rt} = 1$, where the subscript *rt* indicates round trip. A scaling factor of 0.03/70 is used, along with an offset of 0.0254 s. Thus, the modeled distribution in Fig. 11 comes from a random variable of the form

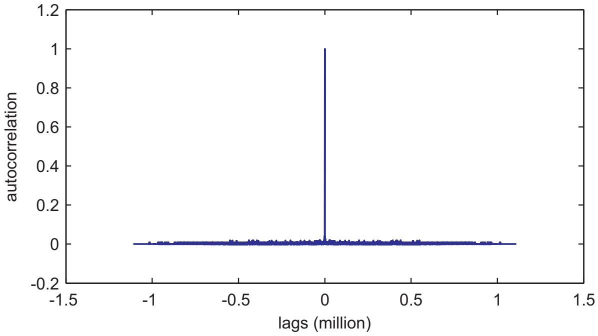

FIG. 10. AUTOCORRELATION OF DELAY.

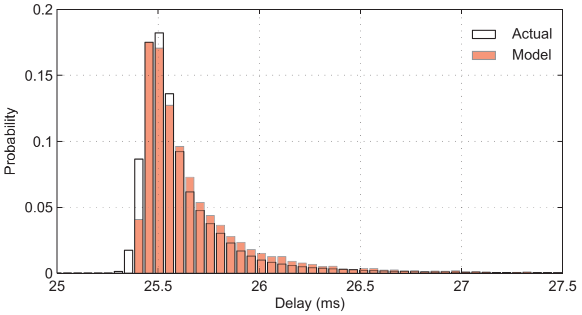

FIG. 11. ACTUAL VS. MODELED DISTRIBUTION OF DELAY BETWEEN ANN ARBOR AND WARREN.

$$X_{rt} \sim 0.0254 + \frac{0.03}{70} \text{Log-N}\left(\mu_{rt} = -0.9, \sigma_{rt} = 1\right) \qquad (14)$$

The model given in Eq. (14) is for the round trip time delay. However, in the simulation, a model for the one-way delay is needed. To find the parameters for the one-way delay model, first recall that for two independent random variables the mean of their sum is the sum of their means, and the variance of their sum or difference is the sum of their variances. Second, recall that the mean of a lognormal distribution is given by $e^{\mu+\sigma^2/2}$, and the variance is $e^{2\mu+\sigma^2}(e^{\sigma^2}-1)$. Finally, assume that the network characteristics are the same both ways. Then, the parameters μ_{ow} and σ_{ow} for the one-way delay model can be found by solving

$$\begin{aligned} e^{\mu_{ow}+\sigma_{ow}^2/2} &= \frac{1}{2} e^{\mu_{rt}+\sigma_{rt}^2/2} \\ e^{2\mu_{ow}+\sigma_{ow}^2}\left(e^{\sigma_{ow}^2}-1\right) &= \frac{1}{2} e^{2\mu_{rt}+\sigma_{rt}^2}\left(e^{\sigma_{rt}^2}-1\right) \end{aligned} \qquad (15)$$

This results in the following one-way delay model

$$\begin{aligned} X_{ow} \sim & \frac{0.0254}{2} \\ & + \frac{0.03}{70} \text{Log-N}\left(\mu_{ow} = -1.838, \sigma_{ow} = 1.221\right) \end{aligned} \qquad (16)$$

and the sum of two such one-way delays results in the desired distribution for the round trip delay.

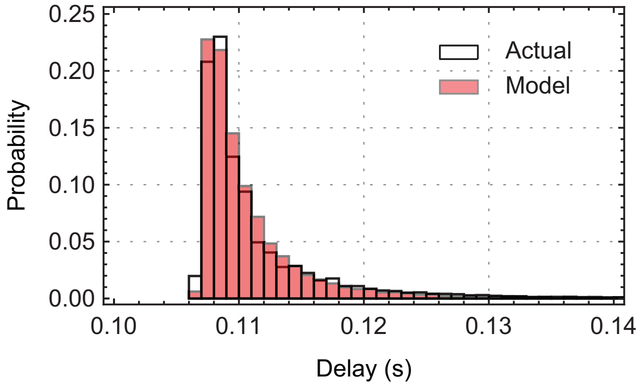

FIG. 12. ACTUAL VS. MODELED DISTRIBUTION OF DELAY BETWEEN MICHIGAN AND CALIFORNIA.

To obtain a model that represents a longer connection, the network characterization given in detail above for an Ann Arbor – Warren connection was repeated for a connection between Michigan and California. The actual and model-based distributions are given in Fig. 12. The one-way delay model in this case becomes

$$X_{ow} \sim \frac{0.1068}{2} + \frac{1.083}{175}\text{Log-N}\left(\mu_{ow} = -1.838, \sigma_{ow} = 1.221\right) \tag{17}$$

Simulation Results

The ILC framework shown in Fig. 2 was used independently for all the coupling variables, i.e., the throttle, shaft torque, and shaft speed. All three ILC controllers independently used the learning algorithm

$$u_i^{m+1}(k) = u_i^m(k) + k_p e_i^m(k) + k_d\left(e_i^m(k) - e_i^m(k-1)\right) \tag{18}$$

with k_p and k_d being control parameters that can be tuned independently for each controller. Thus, for this case study, the ILC framework variables can be defined as in Table 2. The ILC action for shaft torque is injected on the System 1 side of the Internet, whereas the ILC actions for shaft speed and throttle are injected on the System 2 side.

First set of simulations were performed for the first 130 seconds of the FTP75 drive cycle. The control parameters were tuned for each delay model separately using a genetic optimizer that sought to minimize the system error as given in Eq. (2) after 10 ILC iterations. The resulting parameters are shown in Table 3.

TABLE 2. VARIABLES IN THE ILC FRAMEWORK FOR THE CASE STUDY

Variable	Description
$c_{1,j}$	Shaft torque on System j side, $j = 1,2$
e_1	Error in shaft torque, $e_1 = c_{1,2} - c_{1,1}$
$c_{2,j}$	Shaft speed on System j side, $j = 1,2$
e_2	Error in shaft torque, $e_2 = c_{2,1} - c_{2,2}$
$c_{3,j}$	Throttle on System j side, $j = 1,2$
e_3	Error in throttle, $e_3 = c_{3,1} - c_{3,2}$

TABLE 3. ILC PARAMETERS USED IN THE CASE STUDY

Coupling variable	ILC parameters for Ann Arbor – Warren	ILC parameters for Michigan - California
Torque	$k_p = 0.697;\ k_d = 0.496$	$k_p = 0.197;\ k_d = 0.084$
Speed	$k_p = 0.735;\ k_d = 0.138$	$k_p = 0.274;\ k_d = 0.005$
Throttle	$k_p = 0.719;\ k_d = 0.004$	$k_p = 0.503;\ k_d = 0.031$

A representative result for the Ann Arbor – Warren delay scenario is summarized in Fig. 13, which shows the ILC performance of the three coupling variables for 10 ILC iterations. The figure is representative in the sense that each time it is re-generated, the curves would look slightly different due to the random nature of delay. Nevertheless, the performance level stays the same. Specifically, the figure shows a dramatic reduction in the errors in coupling variables. More than 95% reduction is achieved in the normalized 2-norms of errors of all coupling variables, where errors are normalized with respect to the 2-norms of the corresponding initial errors

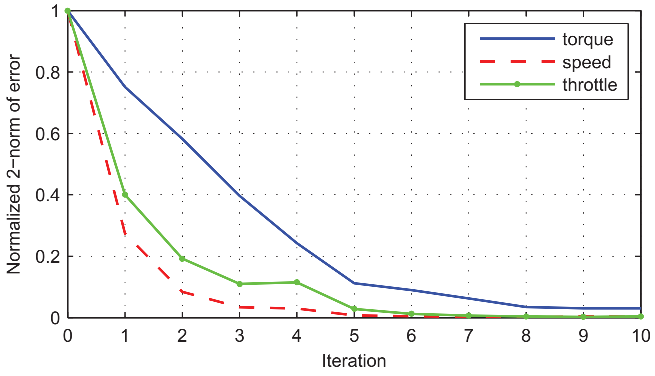

FIG. 13. REPRESENTATIVE ILC PERFORMANCE FOR THE THREE COUPLING VARIABLES FOR THE ANN ARBOR – WARREN DELAY. LOWER ERROR VALUES MEAN HIGHER FIDELITY.

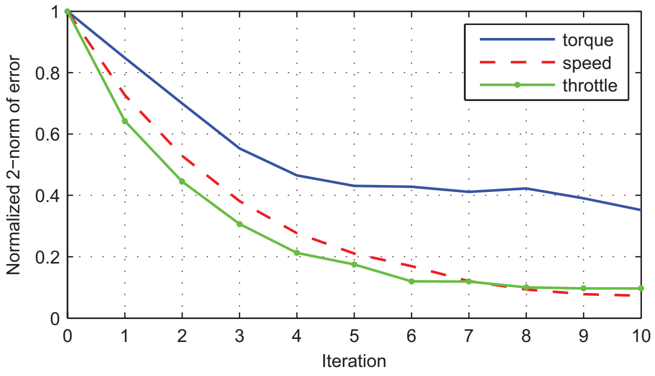

FIG. 14. REPRESENTATIVE ILC PERFORMANCE FOR THE THREE COUPLING VARIABLES FOR THE MICHIGAN – CALIFORNIA DELAY. LOWER ERROR VALUES MEAN HIGHER FIDELITY.

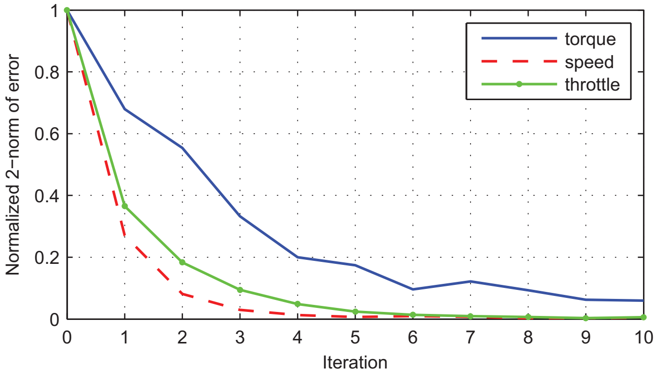

FIG. 15. REPRESENTATIVE ILC PERFORMANCE FOR THE 130-340 SEC WINDOW OF THE FTP75 AND FOR THE ANN ARBOR – WARREN DELAY.

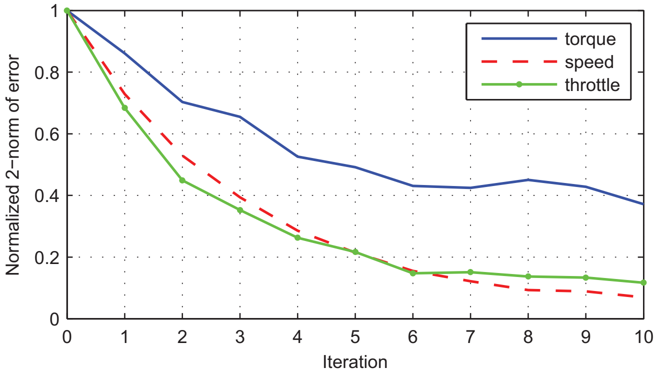

FIG. 16. REPRESENTATIVE ILC PERFORMANCE FOR THE 130-340 SEC WINDOW OF THE FTP75 AND FOR THE MICHIGAN – CALIFORNIA DELAY.

as given in Eq. (3).

Figure 14 shows a representative ILC performance for the MI – CA delay model. As a result of the larger variation in delay, the performance in terms of reducing the error within 10 iterations is not as good as in the previous case. Nevertheless, the ILC is still capable of reducing the error by around 90% for speed and throttle, and around 60% for torque. This result also highlights the fact that different coupling variables may present different levels of challenge for improving their fidelity, a result consistent with previously published results [22, 23, 32, 33]. In this particular scenario, the torque signal is more sensitive to delay than the speed and throttle signals, since the speed signal is filtered through vehicle inertia and the throttle signal is closely related to speed. Hence, it is harder to improve the fidelity of the torque signal.

A comparison of the ILC parameters for the two delay conditions shows that the optimization reduces the proportional gains for all variables in the MI – CA delay scenario to cope with the increased variation in delay through a slower learning. However, it is interesting that the reduction in proportional gain is not to the same extent for the throttle signal as for the torque and speed signals. This again emphasizes the different characteristics different coupling variables may exhibit.

Figures 15 and 16 show representative ILC performances for a different drive cycle than what was used for ILC parameter tuning. Specifically, the 130s-340s time window of the FTP75 was used in this part of the study, which exhibits a more aggressive driving behavior than the 0s-130s time window. The figures show that the ILC performance levels observed for the tuning drive cycle are still maintained. Thus, the proposed ILC framework seems to be robust to the drive cycle.

These results highlight the merit and potential of the proposed framework to improve fidelity, and encourage the further development of this framework. Some specific questions of interest are what the system requirements are for the proposed framework to work, what type of learning functions are best suited for this application, how the ILC parameters relate to system characteristics, and what the robustness of this approach is and how it can be maximized. In terms of robustness, the robustness to variations in the initial conditions when actual hardware is introduced and the robustness to variations in the network delays are of particular importance. Potential future work also includes a centralized ILC approach, as opposed to the decentralized approached used in this study, where every coupling variable has its independent controller. A centralized approach could yield a different performance in terms of final error, convergence speed, or robustness, which may, however, come at the expense of ease of scalability.

V. SUMMARY AND CONCLUSIONS

An Iterative Learning Control based framework has been developed to improve fidelity in Internet-distributed hardware-in-the-loop simulation. The framework has been tested using a purely simulation-based case study that showed that 60-95% reduction in coupling variable errors can be obtained with the proposed method. The results imply that the framework can improve the fidelity of the networked simulation significantly, and encourage further development and experimentation with actual hardware in the loop.

ACKNOWLEDGMENTS

This work was supported by the Automotive Research Center (ARC), a U.S. Army Center of Excellence in Modeling and Simulation of Ground Vehicles.

REFERENCES

[1] Fathy, H. K., Filipi, Z. S., Hagena, J., and Stein, J. L., 2006, "Review of Hardware-in-the-Loop Simulation and Its Prospects in the Automotive Area", *Proceedings of SPIE - Modeling and Simulation for Military Applications*, Kissimmee, FL, United States, Apr 18-21, 2006, SPIE, **6228**, pp. 1-20.

[2] Kimura, A. and Maeda, I., 1996, "Development of Engine Control System Using Real Time Simulator", *Proceedings of 1996 IEEE International Symposium on Computer-Aided Control System Design*, Sep 15-18 1996, pp. 157-163.

[3] Verma, R., Del Vecchio, D., and Fathy, H. K., 2008, "Development of a Scaled Vehicle with Longitudinal Dynamics of an HMMWV for an ITS Testbed", IEEE/ASME Transactions on Mechatronics, **13**(1), pp. 46-57.

[4] Leitner, J., 2001, "A Hardware-in-the-Loop Testbed for Spacecraft Formation Flying Applications", *Proceedings of 2001 IEEE Aerospace Conference*, 10-17 March 2001, IEEE, **2**, pp. 615-620.

[5] Yue, X., Vilathgamuwa, D. M., and Tseng, K.-J., 2005, "Robust Adaptive Control of a Three-Axis Motion Simulator with State Observers", IEEE/ASME Transactions on Mechatronics, **10**(4), pp. 437-448.

[6] Ganguli, A., Deraemaeker, A., Horodinca, M., and Preumont, A., 2005, "Active Damping of Chatter in Machine Tools - Demonstration with a 'Hardware-in-the-Loop' Simulator", Journal of Systems and Control Engineering, **219**(5), pp. 359-369.

[7] Aghili, F. and Piedboeuf, J.-C., 2002, "Contact Dynamics Emulation for Hardware-in-Loop Simulation of Robots Interacting with Environment", *Proceedings of 2002 IEEE International Conference on Robotics and Automation*, Washington, D.C., May 11-15 2002, IEEE, **1**, pp. 523-529.

[8] White, G. D., Bhatt, R. M., Tang, C. P., and Krovi, V. N., 2009, "Experimental Evaluation of Dynamic Redundancy Resolution in a Nonholonomic Wheeled Mobile Manipulator", IEEE/ASME Transactions on Mechatronics, **14**(3), pp. 349-357.

[9] Buford, J. A., Jr., Jolly, A. C., Mobley, S. B., and Sholes, W. J., 2000, "Advancements in Hardware-in-the-Loop Simulations at the U.S. Army Aviation and Missile Command", *Proceedings of SPIE - Technologies for Synthetic Environments: Hardware-in-the-Loop Testing V*, 24-26 April 2000, R. L. Murrer, ed., SPIE, **4027**, pp. 2-10.

[10] Huber Jr, E. G. and Courtney, R. A., 1997, "Hardware-in-the-Loop Simulation at Wright Laboratory's Dynamic Infrared Missile Evaluator (DIME) Facility", *Proceedings of 1997 Technologies for Synthetic Environments: Hardware-in-the-Loop Testing II*, April 21, 1997 - April 23, 1997, SPIE, **3084**, pp. 2-8.

[11] Mahin, S., Nigbor, R., Pancake, C., Reitherman, R., and Wood, S., 2003, "The Establishment of the Nees Consortium", *Proceedings of 2003 ASCE/SEI Structures Congress and Exposition: Engineering Smarter*, May 29, 2003 - May 31, 2003, American Society of Civil Engineers, pp. 181-182.

[12] Spencer, B. F., Elnashai, A., Nakata, N., Saliem, H., Yang, G., Futrelle, J., Glick, W., Marcusiu, D., Ricker, K., Finholt, T., Horn, D., Hubbard, P., Keahey, K., Liming, L., Zaluzec, N., Pearlman, L., and Stauffer, E., 2004, "The Most Experiment: Earthquake Engineering on the Grid", Report Technical Report NEESgrid-2004-41, NEESgrid.

[13] Pan, P., Tada, M., and Nakashima, M., 2005, "Online Hybrid Test by Internet Linkage of Distributed Test-Analysis Domains", Earthquake Engineering and Structural Dynamics, **34**(11), pp. 1407-1425.

[14] Stojadinovic, B., Mosqueda, G., and Mahin, S. A., 2006, "Event-Driven Control System for Geographically Distributed Hybrid Simulation", Journal of Structural Engineering, **132**(1), pp. 68-77.

[15] Takahashi, Y. and Fenves, G. L., 2006, "Software Framework for Distributed Experimental-Computational Simulation of Structural Systems", Earthquake Engineering and Structural Dynamics, **35**(3), pp. 267-291.

[16] Mosqueda, G., Stojadinovic, B., Hanley, J., Sivaselvan, M., and Reinhorn, A. M., 2008, "Hybrid Seismic Response Simulation on a Geographically Distributed Bridge Model", Journal of Structural Engineering, **134**(4), pp. 535-543.

[17] Compere, M., Goodell, J., Simon, M., Smith, W., and Brudnak, M., 2006, "Robust Control Techniques Enabling Duty Cycle Experiments Utilizing a 6-DOF Crewstation Motion Base, a Full Scale Combat Hybrid Electric Power System, and Long Distance Internet Communications", SAE Technical Paper, 2006-01-3077.

[18] Goodell, J., Compere, M., Simon, M., Smith, W., Wright, R., and Brudnak, M., 2006, "Robust Control Techniques for State Tracking in the Presence of Variable Time Delays", SAE Technical Paper, 2006-01-1163.

[19] Brudnak, M., Pozolo, M., Paul, V., Mohammad, S., Smith, W., Compere, M., Goodell, J., Holtz, D., Mortsfield, T., and Shvartsman, A., 2007, "Soldier/Harware-in-the-Loop Simulation-Based Combat Vehicle Duty Cycle Measurement: Duty Cycle Experiment 2", *Proceedings of Simulation Interoperability Workshop*, Norfolk, VA, March 2007, Simulation Interoperability Standards Organization (SISO), SIW-07S-016.

[20] Ersal, T., Brudnak, M., Stein, J. L., and Fathy, H. K., 2009, "Variation-Based Transparency Analysis of an Internet-Distributed Hardware-in-the-Loop Simulation Platform for Vehicle Powertrain Systems", *Proceedings of ASME Dynamic Systems and Control Conference*, Hollywood, California, October 12-14, 2009, ASME.

[21] Ersal, T., Brudnak, M., Salvi, A., Stein, J. L., Filipi, Z., and Fathy, H. K., 2009, "Development of an Internet-Distributed Hardware-in-the-Loop Simulation Platform for an Automotive Application", *Proceedings of ASME Dynamic Systems and Control Conference*, Hollywood, California, October 12-14, 2009, ASME.

[22] Ersal, T., Brudnak, M., Salvi, A., Stein, J. L., Filipi, Z., and Fathy, H. K., 2011, "Development and Model-Based Transparency Analysis of an Internet-Distributed Hardware-in-the-Loop Simulation Platform", Mechatronics, **21**(1), pp. 22-29, doi: 10.1016/j.mechatronics.2010.08.002.

[23] Ersal, T., Brudnak, M., Stein, J. L., and Fathy, H. K., 2012, "Statistical Transparency Analysis in Internet-Distributed Hardware-in-the-Loop Simulation", IEEE/ASME Transactions on Mechatronics, **17**(2), pp. 228-238 doi: 10.1109/TMECH.2010.2095024.

[24] Kress, R. L., Hamel, W. R., Murray, P., and Bills, K., 2001, "Control Strategies for Teleoperated Internet Assembly", IEEE/ASME Transactions on Mechatronics, **6**(4), pp. 410-416.

[25] Elhajj, I., Tan, J., Xi, N., Fung, W. K., Liu, Y. H., Kaga, T., Hasegawa, Y., and Fukuda, T., 2002, "Multi-Site Internet-Based Tele-Cooperation", Integrated Computer-Aided Engineering, **9**(2), pp. 117-127.

[26] Munir, S. and Book, W. J., 2002, "Internet-Based Teleoperation Using Wave Variables with Prediction", IEEE/ASME Transactions on Mechatronics, **7**(2), pp. 124-133.

[27] Niemeyer, G. and Slotine, J.-J. E., "Toward Bilateral Internet Teleoperation", in *Beyond Webcams: An Introduction to Online Robots*, MIT Press, 2002, pp. 193-213.

[28] Sun, L.-N., Xie, X.-H., Fu, L.-X., and Du, Z.-J., 2003, "Internet-Based Telerobotic Surgery: Problems and Approaches", Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology, **35**(2), pp. 129-133.

[29] Shi, Y.-H. and Wang, Y.-C., 2004, "Study on Internet-Based Force Feedback Technology", Robot, **26**(4), pp. 330-335.

[30] Slawinski, E., Postigo, J. F., and Mut, V., 2007, "Bilateral Teleoperation through the Internet", Robotics and Autonomous Systems, **55**(3), pp. 205-215, doi: 10.1016/j.robot.2006.09.002.

[31] Chopra, N., Berestesky, P., and Spong, M. W., 2008, "Bilateral Teleoperation over Unreliable Communication Networks", IEEE

Transactions on Control Systems Technology, **16**(2), pp. 304-13, doi: 10.1109/tcst.2007.903397.

[32] Ersal, T., Gillespie, R. B., Brudnak, M., Stein, J. L., and Fathy, H. K., in press, "Effect of Coupling Point Selection on Distortion in Internet-Distributed Hardware-in-the-Loop Simulation", International Journal of Vehicle Design, **Special Issue on Modeling and Simulation of Ground Vehicle Systems**.

[33] Ersal, T., Gillespie, R. B., Brudnak, M., Stein, J. L., and Fathy, H. K., 2011, "Effect of Coupling Point Selection on Distortion in Internet-Distributed Hardware-in-the-Loop Simulation", *Proceedings of American Control Conference*, San Francisco, CA, USA.

[34] Lawrence, D. A., 1993, "Stability and Transparency in Bilateral Teleoperation", IEEE Transactions on Robotics and Automation, **9**(5), pp. 624-637.

[35] Hashtrudi-Zaad, K. and Salcudean, S. E., 2002, "Transparency in Time-Delayed Systems and the Effect of Local Force Feedback for Transparent Teleoperation", IEEE Transactions on Robotics and Automation, **18**(1), pp. 108-114.

[36] Fite, K. B., Speich, J. E., and Goldfarb, M., 2001, "Transparency and Stability Robustness in Two-Channel Bilateral Telemanipulation", Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control, **123**(3), pp. 400-7.

[37] Çavuşoğlu, M. C., Sherman, A., and Tendick, F., 2002, "Design of Bilateral Teleoperation Controllers for Haptic Exploration and Telemanipulation of Soft Environments", IEEE Transactions on Robotics and Automation, **18**(4), pp. 641-647.

[38] De Gersem, G., Van Brussel, H., and Tendick, F., 2005, "Reliable and Enhanced Stiffness Perception in Soft-Tissue Telemanipulation", International Journal of Robotics Research, **24**(10), pp. 805-822.

[39] Yokokohji, Y. and Yoshikawa, T., 1994, "Bilateral Control of Master-Slave Manipulators for Ideal Kinesthetic Coupling - Formulation and Experiment", IEEE Transactions on Robotics and Automation, **10**(5), pp. 605-619.

[40] Yokokohji, Y., Imaida, T., and Yoshikawa, T., 1999, "Bilateral Teleoperation under Time-Varying Communication Delay", *Proceedings of 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99): Human and Environment Friendly Robots with High Intelligence and Emotional Quotients'*, October 17, 1999 - October 21, 1999, IEEE, **3**, pp. 1854-1859.

[41] Griffiths, P. G., Gillespie, R. B., and Freudenberg, J. S., 2011, "A Fundamental Linear Systems Conflict between Performance and Passivity in Haptic Rendering", IEEE Transactions on Robotics, **27**(1), pp. 75-88.

[42] Griffiths, P. G., Gillespie, R. B., and Freudenberg, J. S., 2008, "A Fundamental Tradeoff between Performance and Sensitivity within Haptic Rendering", IEEE Transactions on Robotics, **24**(3), pp. 537-548.

[43] Freudenberg, J. S., Hollot, C. V., Middleton, R. H., and Toochinda, V., 2003, "Fundamental Design Limitations of the General Control Configuration", IEEE Transactions on Automatic Control, **48**(8), pp. 1355-1370.

[44] Bristow, D. A., Tharayil, M., and Alleyne, A. G., 2006, "A Survey of Iterative Learning Control: A Learning-Based Method for High-Performance Tracking Control", IEEE Control Systems Magazine, **26**(3), pp. 96-114.

[45] Xu, J.-X. and Tan, Y., 2002, "Robust Optimal Design and Convergence Properties Analysis of Iterative Learning Control Approaches", Automatica, **38**(11), pp. 1867-1880.

[46] Wang, D., 2000, "On D-Type and P-Type Ilc Designs and Anticipatory Approach", International Journal of Control, **73**(10), pp. 890-901.

[47] Cheah, C.-C. and Wang, D., 1998, "Learning Impedance Control for Robotic Manipulators", IEEE Transactions on Robotics and Automation, **14**(3), pp. 452-465.

[48] Chien, C.-J. and Liu, J.-S., 1996, "A P-Type Iterative Learning Controller for Robust Output Tracking of Nonlinear Time-Varying Systems", International Journal of Control, **64**(2), pp. 319-334.

[49] Heinzinger, G., Fenwick, D., Paden, B., and Miyazaki, F., 1992, "Stability of Learning Control with Disturbances and Uncertain Initial Conditions", IEEE Transactions on Automatic Control, **37**(1), pp. 110-114.

[50] Park, K.-H., Bien, Z., and Hwang, D.-H., 1999, "A Study on the Robustness of a Pid-Type Iterative Learning Controller against Initial State Error", International Journal of Systems Science, **30**(1), pp. 49-59.

[51] Saab, S. S., 2003, "Stochastic P-Type/D-Type Iterative Learning Control Algorithms", International Journal of Control, **76**(2), pp. 139-148.

[52] Horowitz, R., 1993, "Learning Control of Robot Manipulators", Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, **115**(2 B), pp. 402-411.

[53] Havlicsek, H. and Alleyne, A., 1999, "Nonlinear Control of an Electrohydraulic Injection Molding Machine Via Iterative Adaptive Learning", IEEE/ASME Transactions on Mechatronics, **4**(3), pp. 312-323.

[54] Arimoto, S., Kawamura, S., and Miyazaki, F., 1984, "Bettering Operation of Robots by Learning", Journal of Robotic Systems, **1**(2), pp. 123-140.